Title	モジュールの再構成 : 共有を許す場合 (情報科学の数学的基礎理論と応用)
Author(s)	菊野, 亨; 竹中, 春之
Citation	数理解析研究所講究録 (1979), 353: 97-106
Issue Date	1979-04
URL	http://hdl.handle.net/2433/104423
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

数理解析研究所講究録
353巻 1979年 97-106

モジュールの再構成
——共有を許す場合——

広島大学 工学部 菊野 亨
竹中 春之

1. まえがき

プログラム設計においてプログラムの品質(例えば, 信頼性, 変更の容易さなど)を高めることは重要な問題の1つであり, いわゆる"structured programming"に見られる様に, プログラムの構造の良さに対する関心は深い[1]. 特に, プログラム設計の初期段階においてはプログラムの構造の良さに関する十分な検討が必要である. 設計の初期段階では, 通常, プログラムはコール関係のグラフGで表される. Stevensら[1]は, モジュールm_iの判定結果が他のモジュールに影響を与えるとき, その影響を受けるモジュールの集合(エフェクトの範囲)が, m_iの子孫のモジュール† の集合(制御の範囲)に完全に含まれることが望ましいと考え, コ

† m_iが次々とコールするモジュールのこと

ール関係のグラフGに対し，①モジュールの合併，及び，②コール関係の変更の操作を行い，エフェクトの範囲が制御の範囲に含まれる様にモジュールを再構成することを提案している．

一方，葛山ら[(2)]は，Stevensらの方法の場合モジュールが大きくなりすぎるという欠点をもつことを指摘し，その解決策として，Gを有向木に限って，Gに対しモジュールの合併だけを行う再構成法を提案している．その際，m_iのエフェクトの範囲をm_iの子孫のモジュールだけに限らず，親のモジュールと兄弟のモジュールにまで許す様にしている．

本論文では，葛山らの結果[(2)]に対し若干の拡張を行う．つまり，モジュールの共有が記述できる様にGを拡張した場合について，文献（2）と基本的に同じ立場で再構成の問題を考察する．

2．準備

2.1　諸定義

有向木$T=(M, A, m_0)$を通常の通り定義する[(3)]．ここで，$M=\{m_i\}$は節点の集合，Aは有向枝の有限集合，$m_0 \in M$はTの頂点である．有向木Tの葉のある部分集合をL（$L \subseteq M$）とし，$\bar{L}=M-L$とする．

［定義１］　上述のT，L，$\bar{L}$に関して“各$m_\ell \in L$に対し，$\bar{L}$に属する節点からm_ℓに向かう有向枝の追加を許して”Tから得られる有向グラフを$G=(M, C, m_0)$と表し，S木と呼ぶ．特に，S木Gが次の条件Cを満たすとき，Gをプログラムモジュール間のコール関係のグラフと呼ぶ．便宜上，コール関係のグラフGの節点m_iをモジュールm_iとも呼ぶ．

条件C：structure chart[1]におけるモジュールm_iにGの節点$m_i \in M$が対応し，モジュールm_iからモジュールm_jへのコール関係にGの有向枝$(m_i, m_j) \in C$が対応している．

以降では，Gはコール関係のグラフを表すものとする．Gにおいて，m_iからm_jへ有向枝があることを$m_i \longrightarrow m_j$と書く．又，記法$m_i \xrightarrow{*} m_j$によってm_iからm_jへ有向枝の向きに沿った道が存在することを表す．$m_i \xrightarrow{*} m_j$の道の長さが１以上のとき，特に$m_i \xrightarrow{+} m_j$で表す．なお，逆に有向枝や道が存在しない場合には，$m_i \not\longrightarrow m_j$，$m_i \not\xrightarrow{*} m_j$などと表す．Gにおいて，$m_i \longrightarrow m_j$が成り立つとする．このとき$m_i$を$m_j$の親といい，更に$m_i \longrightarrow m_k$（$k \neq j$）が成立するなら$m_k$を$m_j$の兄弟という．又，Gにおいて$m_j \xrightarrow{*} m_\ell$が成り立つなら$m_\ell$を$m_j$の子孫という．

［定義２］　$G=(M, C, m_0)$の節点の集合Mに対し，Mのある分割を$\pi=\{\beta_1, \beta_2, \cdots, \beta_\ell\}$と表す．各$\beta_i$をブ

ロックと呼ぶ．$\beta_i \neq \beta_j$で，且つ，$m_k \longrightarrow m_\ell$（$m_k \in \beta_i$，$m_\ell \in \beta_j$）なる２つの節点$m_k$，$m_\ell$が存在するとき，$\beta_i \Rightarrow \beta_j$と表す．

［定義３］　G上のモジュールm_iとm_jに対し，m_i内のある判定の結果がm_j内におけるコントロールの流れに影響を与える時，その判定からm_jへエフェクトがあるという．そのとき，節点対（m_i，m_j）をエフェクト枝と呼び，エフェクト枝の集合をEで表す．

［定義４］　$G=(M, C, m_0)$とエフェクト枝(m_i, m_j)に対し，次の条件①，②を満たす節点m_kをm_i，m_jの"lowest common ancestor"と呼び，$lca(m_i, m_j)$で表す．その集合を$LCA(m_i, m_j)$と書く．

① $m_k \xrightarrow{*} m_i$，且つ，$m_k \xrightarrow{*} m_j$

② 次のⓐ，ⓑの少なくとも１つが成り立つ．

ⓐ道$m_k \xrightarrow{*} m_i$が存在し，その道上の任意の節点m_ℓ（$\ell \neq i, k$）について$m_\ell \not\xrightarrow{*} m_j$である．

ⓑ道$m_k \xrightarrow{*} m_j$が存在し，その道上の任意の節点m_ℓ（$\ell \neq j, k$）について$m_\ell \not\xrightarrow{*} m_i$である．

図１のグラフの実線部分にコール関係のグラフの例を示す．又，同図で点線で描く有向枝はエフェクト枝を示す．図１で$LCA(m_7, m_3) = \{m_2, m_4\}$となる．

2.2　問題

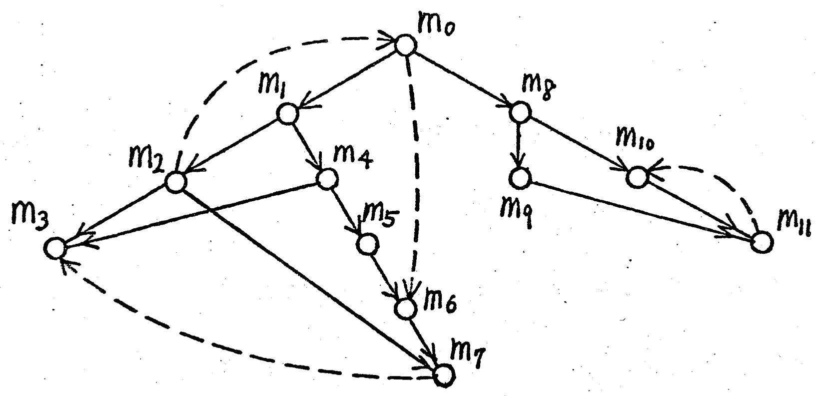

図1　コール関係のグラフとエフェクト枝

以上の準備のもとに，本稿で検討するプログラムモジュールのエフェクト関係を考慮したモジュールの再構成問題を，文献（2）に従って述べる．

［問題1］　G＝（M，C，m_0）と集合Eが与えられたとき，次の3つの条件を満足するブロック数最大の分割$\pi = \{\beta_1, \beta_2, \cdots, \beta_l\}$を1つ求めよ．

［連結条件］　各ブロックβ_iに属する任意の2つの節点に対し，節点間の有向枝を無向枝とみなしたとき，一方の節点から他方の節点へ至る道が存在する．

［非サイクリック条件］　関係⇒によって閉路（$\beta_i \Rightarrow \cdots \Rightarrow \beta_i$）をつくることはない．

［近接条件］　Eに属する各エフェクト枝（m_i，m_j）に対し，m_i，m_jを含むブロックをβ_s，β_tとする．このときLCA（m_i，m_j）に属する各$lca(m_i, m_j)$に対し次の条件①，②が成り立つ．なお$lca(m_i, m_j)$を含むブロックをβ_uと仮定

する．

①次の（a）～（b）の少なくとも1つが成立する．

（a）　$\beta_u = \beta_s = \beta_t$

（b）　$\beta_u = \beta_s$，且つ，$\beta_u \Rightarrow \beta_t$

（c）　$\beta_u = \beta_t$，且つ，$\beta_u \Rightarrow \beta_s$

（d）　$\beta_u \Rightarrow \beta_s$，且つ，$\beta_u \Rightarrow \beta_t$

②任意の$m_k \in \beta_s$に対し，$lca(m_i, m_j) \xrightarrow{+} m \xrightarrow{+} m_k$なる$m \notin \beta_s \cup \beta_u$は存在せず，任意の$m_k \in \beta_t$に対し，$lca(m_i, m_j) \xrightarrow{+} m \xrightarrow{+} m_k$なる$m \notin \beta_t \cup \beta_u$は存在しない．

3．アルゴリズム

3.1　カット枝集合を求める問題への変換

Gに対し〔連結条件〕を満足している分割πを指定するには，ブロック間の枝の集合を与えてもよい．このような枝を“カット枝”と呼ぶ．

$G = (M, C, m_0)$とEに対し，以下に示す変換τを適用して，問題1を後述の問題2に変換する．

G上でk（$k \geq 2$）本の枝が入っている節点m_iに対し，k個のコピー$m_{i(1)}, \cdots, m_{i(k)}$を作り，Gを有向木$\tau(G)$に変換する．Eに属するエフェクト枝$(m_i, m_j)$に対しては，$m_i$，$m_j$（及び$m_{i(p)}$，$m_{j(p)}$）から，$lca(m_i, m_j)$に向かうエフェ

クト枝を新らたに作る．こうして得られるエフェクト枝の集合をτ(E)と表す．図1のGとEに対し，変換τを適用すると図2に示すτ(G)とτ(E)が得られる．G及びτ(G)において，そこから出る枝の数が0である節点をΔ節点と呼ぶ．又，Δ節点に入る枝をΔ枝と呼ぶ．

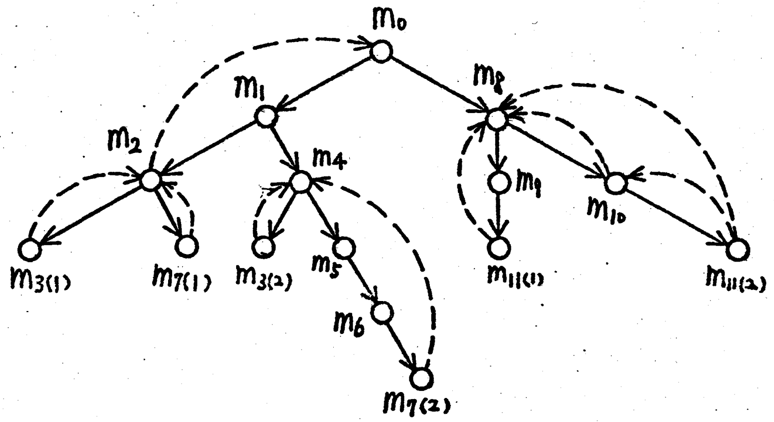

図2　変換τ

Eに属する任意のエフェクト枝(m_i, m_j)に対し$m_j \xrightarrow{*} m_i$が成り立つようなEのクラスを$\mathcal{E}$と書く．

［問題2］　有向木$\hat{G}$と$\mathcal{E}$に属するエフェクト関係Bが与えられたとき，以下で述べる［カット条件］を満足するカット枝の集合Δで，枝の個数が最大のものを1つ求めよ．

［カット条件］　B内の任意のエフェクト枝(m_i, m_j)に対し，$\hat{G}$上の道$m_j \xrightarrow{*} m_i$がΔ内の枝を高々1個しか含まない．

GとEに対する問題1が$\hat{G}=\tau(G)$，$B=\tau(E)$とおいた

問題2に帰着されることを示す．

［定理1］ GとEが与えられた時，τ(G)とτ(E)に対し，［カット条件］を満足する枝の集合で，すべてのΔ枝を含む枝数最大のものを$\bar{\Delta}$とすると，$\bar{\Delta}$で決まる分割πは，GとEに対し，［連結条件］［近接条件］［非サイクリック条件］を満足するGの節点の分割で，しかもΔ節点を1つのブロックとするブロック数最大のものである．

問題2の解Δ（定理1の$\bar{\Delta}$）を求めるアルゴリズムは文献（2）で既に与えられている．図2のτ(G)とτ(E)に対し，［Δを求める手続き］(2)を適用すると，$\Delta=\{(m_0, m_8), (m_1, m_2), (m_1, m_4), (m_2, m_{3(1)}), (m_2, m_{7(1)}), (m_4, m_{3(2)}), (m_6, m_{7(2)}), (m_9, m_{11(1)}), (m_{10}, m_{11(2)})\}$と求まる．図3では$\Delta$に属する枝を太線で描いている．

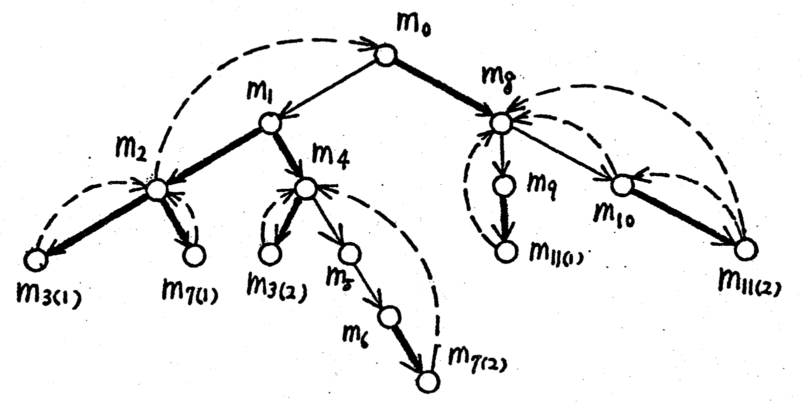

図3 ［カット条件］を満たす枝の集合Δ

（有向枝→の部分だけ）

以上まとめると，問題1の解を求めるアルゴリズムMは次の様になる．

アルゴリズムM

［$LCA(m_i, m_j)$の計算］

［変換τ］

［Δを求める手続き］

［分割πの決定］：τ(G)上の枝の集合ΔをG上の（節点に対する）分割に変換する．

アルゴリズムMを図1のGとEに対し適用すると，図2，図3と変換されていって，Δが求まる．そのΔから分割πを求めると，$\pi=\{\beta_1, \beta_2, \cdots, \beta_7\}$となる．但し，$\beta_1=\{m_0, m_1\}$，$\beta_2=\{m_2\}$，$\beta_3=\{m_3\}$，$\beta_4=\{m_4, m_5, m_6\}$，$\beta_5=\{m_7\}$，$\beta_6=\{m_8, m_9, m_{10}\}$，$\beta_7=\{m_{11}\}$である．図4には各ブロックを1つの節点と見做して（コール関係の）グ

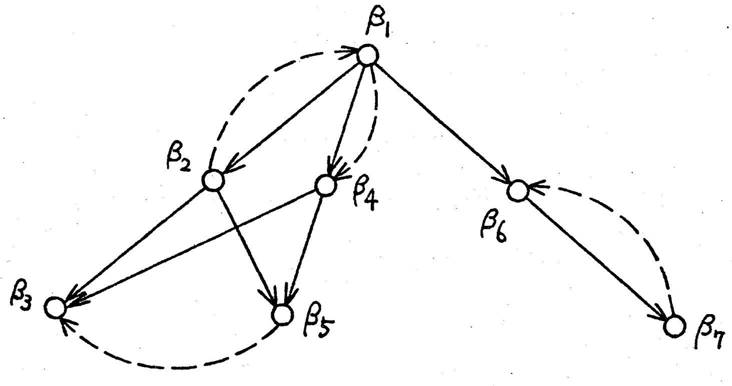

図4　再構成後のモジュール

ラフを示している．図1と図4とを比較すれば，モジュール合併により，（影響関係に関して）構造がかなり良くなっていることがわかる．

［定理2］　与えられた$G=(M, C, m_0)$とEに対し，GがS木であれば，問題1の解であるブロック数最大の分割が$|E| \times |C| \times |M|$のオーダの手数で求められる．

文献（2）のアルゴリズムに対し若干の拡張を行い，ユール関係を表すグラフGが，有向木の葉節点に他の節点からの追加を許したグラフ（定義1のS木）である場合について，再構成の問題を議論した．

文献

(1) W. P. Stevens, G. J. Myers and L. L. Constantine: "Structured design", IBM Syst. J., 13, 2, p. 115 (1974).

(2) 葛山，谷口，嵩："プログラムモジュール群の再構成"，信学論誌（D），J59-D，11，p. 794（1976）．

(3) A. V. Aho, J. E. Hopcroft and J. D. Ullman:"The design and analysis of computer algorithms", Addison-Wesley Publishing Company (1974).